

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Информационные системы

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ С.А. Виденин

подпись инициалы, фамилия

« ____ » _____ 20 ____ г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

Автоматизированная система обучения

Руководитель

подпись, дата

к.ф.-м.н. доцент

должность, ученая степень

И.А. Ефремов

инициалы, фамилия

Выпускник

подпись, дата

А.С. Дубинчик

инициалы, фамилия

Нормоконтролер

подпись, дата

Л.С.Троценко

инициалы, фамилия

Красноярск 2017

РЕФЕРАТ

Выпускная квалификационная работа по теме «Автоматизированная система обучения» содержит 41 страницу текстового документа, 1 приложение, 7 использованных источника.

ОБУЧЕНИЕ, ИНФОРМАЦИОННАЯ СИСТЕМА, РАЗРАБОТКА САЙТА, ВИРТУАЛЬНАЯ МАШИНА, ВЕБ СЕРВЕР, РОЛИ, ПРОВА ПОЛЬЗОВАТЕЛЕЙ

Объект исследования – школьный сайт.

Цель – создать на виртуальной машине школьный сайт с простым набором функциональности.

Задачи:

- создание сайта с элементами присущими типичной школе, как расписание и дневники;
- виртуализация машины со средой разработки сайта на локальном ПК;
- настройка и установка сервисов как: Apache, MySQL, PHP языка программирования;
- установка системы управления сайтом;
- планирование проекта;
- техническая часть и программная составляющая;
- реализация проекта.

Создание обычного сайта с расписанием и дневниками для школы затрагивает весь спектр современных технологий в информационной области и косвенно, до определённого момента, высокой посещаемости, не касается лишь аппаратно тех. составляющей. Для реализации проекта, необходимо знать Unix команды по настройке платформы и сервисов, разбираться в ИС для выбора или создания одной информационной системы для проекта, необходимо иметь представления о циклах жизни проекта. Нужно знание языка программирования, порой нескольких. Объём литературы в этой области растёт колоссальными темпами.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Общие сведения и предустановка системы	5
1.1 Настройка платформы для реализации проекта	5
1.2 Конфигурация соединения с виртуальной машиной	6
1.3 Конфигурация веб-сервера, СУБД, языка программирования	7
1.4 Установка системы управления сайтами	7
1.5 Drupal 7 и её возможности	8
2 Реализация первой стадии разработки системы обучения	10
2.1 Структура системы	10
2.2 Планирование проекта	11
2.3 Техническая часть	13
2.4 Интерфейс, панель управления обучающей системы	16
3 Реализация проекта	18
3.1 Расписание занятий	18
3.2 Оформление проекта	21
3.3 Использование расписания	22
3.4 Создание дневника	23
3.5 Создание раздела с тестами	24
4 Дополнительные сведения по конфигурации и защите проекта	27
4.1 Настройка FTP соединения, владельца и группы для файлов	27
4.2 Настройка PHP языка программирования	29
4.3 Настройка MySQL	30
4.4 Настройка Apache	33
4.5 Резервное копирование проекта	34
4.6 Firewall или базовые настройки безопасности	37
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	41
ПРИЛОЖЕНИЕ А Техническое задание	42

ВВЕДЕНИЕ

В наше время автоматизированные системы обучения приобретают актуальность. Основные преимущества данного типа систем: общедоступность, возможность удалённого соединения через интернет. Основная направленность, на данный момент, проявляется на школьных порталах в виде электронных дневников, тестов, личных кабинетов для учащихся с домашними заданиями и разделами для родителей где они могут общаться с учителями.

В силу распространённости электронных устройств и все поглощающего интернета, бумажные версии привычных вещей как дневник, учебник и возможно даже тетрадь могут уйти в прошлое.

В работе рассматриваются разделы автоматизированной школьной системы образования: как личный кабинет учащихся, так их учителей и родителей, дневник учащегося, кратко возможности организовывать тесты и система оповещения родителей и учащихся.

Сам сайт реализуется на бесплатной системе управления сайтами, обладающей всеми необходимыми для этого функциями. Надо заметить если не воспользоваться данной разработкой, то реализация проекта не вложится и в годовой период даже группы людей.

1 Общие сведения и предустановка системы

1.1 Настройка платформы для реализации проекта

Разработку сайта можно вести на домашнем компьютере. Операционная система (ОС) физического ПК Windows 10. Далее понадобится виртуальная машина с одной из Unix-подобных систем. Unix системы выбираются для онлайн проектов в силу отсутствия излишних графических интерфейсов, т.е. экономии ресурсов.

Oracle VM VirtualBox как система виртуализации машин. Создание виртуальной машины (ВМ) в данной программе не составляет сложности. Нужно выделить необходимое количество процессорных ядер, VirtualBox распознаёт поддерживает ли процессор виртуализацию и может выделить реальное количество физических ядер процессора. Далее указать количество виртуальной памяти и дискового пространства, на этом создание машины завершается. Установка операционной системы на ВМ возможна через загрузку виртуального диска с ОС с одного из псевдоприводов.

ОС Ubuntu (ОС тип: Linux) 12.04 LTS (32 бита) удобна в установке необходимых компонентов. Нужно установить: Apache 2, PHP 5.3.10, MySQL 5. Надо заметить, что выбор версии Ubuntu сопряжён с версиями пакетов, которые на ней можно установить. Различные системы управления сайтами поддерживают только определённые версии в формате – с...до, например, в нашем случае Drupal 7 работает на PHP 5.3 & MySQL 5.5. Необходимо ещё учитывать, что модули для системы часто запаздывают, и если сама система проходит проверку с более новыми версиями, то модули могут её не пройти.

Предустановка всех необходимых пакетов возможна на стадии установки ОС. Нам необходимо выбрать LAMP:

- Linux – операционная система Linux;
- Apache – веб-сервер;
- MySQL – СУБД;
- PHP – язык программирования, используемый для создания веб-приложений.

В итоге нам будет доступно окно с интерпретатором для работы с ОС рисунок 1.

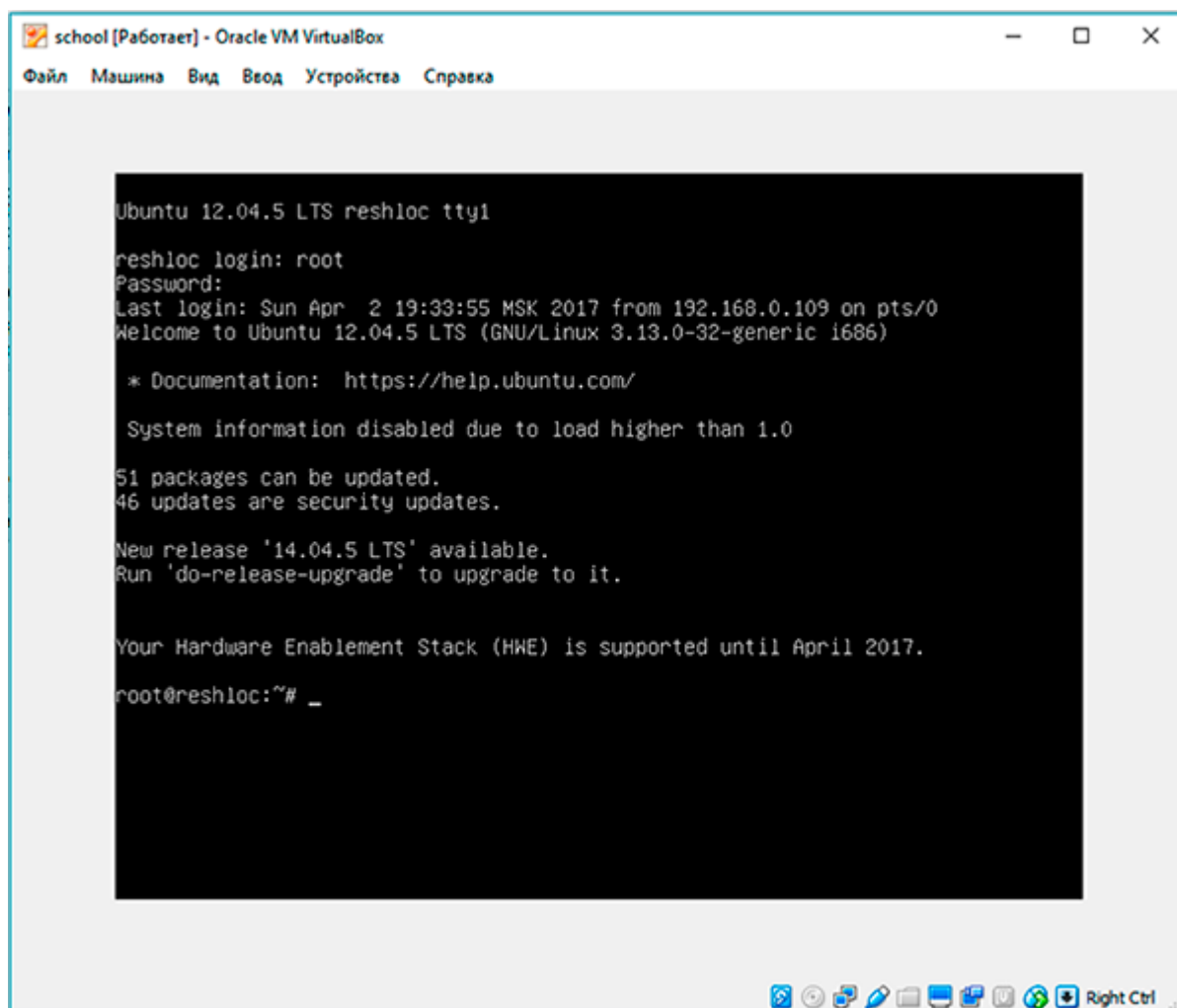


Рисунок 1 – Интерпретатор для работы с ОС

1.2 Конфигурация соединения с виртуальной машиной

По умолчанию при установке Oracle VM VirtualBox в сетевых настройках создаётся VirtualBox Host-Only Ethernet Adapter, который обеспечивает все ВМ сетью. При этом каждой машине присваивается индивидуальный IP домашней сети, что является очень удобным фактором, т.к. ВМ доступна с любого ПК дома.

Следующий шаг для реализации проекта, настроить ОС для открытия сайт в браузере и собственно ведения разработки в каком-либо приложении, в нашем случае Dreamweaver CC. Для браузера надо указать домен сайта и IP куда необходимо ссылаться в host файле Win 10. Часто по политике безопасности ОС данный файл заблокирован на запись.

К ВМ можно подключиться через любой SSH клиент при условии, что мы установили его (ssh) на ВМ. Любой пакет на машине устанавливается *apt-get install название пакета*.

1.3 Конфигурация веб-сервера, СУБД, языка программирования

Настрой сервиса Apache 2 на Ubuntu упрощена, по сравнению с аналогичными ОС. Для того что бы сервис стал обрабатывать запросы, поступающие из вне нужно:

- в директории `/etc/apache2/sites-available` найти файл `default`;
- в файле указать домен и корневую директорию сайта

```
<VirtualHost *:80>
    ServerName school.lc
    ServerAlias www.school.lc
    DocumentRoot /var/www/school
</VirtualHost>;
```

- не забываем установить `mod_rewrite`, `a2enmod rewrite` и перезапускаем сервис.

Можно сказать, что настройки СУБД носят сугубо ограничивающий характер на размер обрабатываемых пакетов и их количество. Однако при правильном использовании можно правильно распределять нагрузку между жёстким диском и оперативной памятью. Вся конфигурация происходит в файле `/etc/mysql/my.cnf`. Коммуникационный пакет – это одна команда SQL, посылаемая серверу, или одна строка, посылаемая клиенту.

Конфигурация PHP очень важна, система управления сайтами не станет работать, если недостаточно памяти выделяемой для процесса (в отдельности), так и времени процессора в доли секундах. При этом необходимо учитывать, что мы установили все доп. библиотеки, которые нам нужны. Конфигуровочный файл находится по адресу `/etc/php5/apache2/php.ini`.

1.4 Установка системы управления сайтами

Необходимо загрузить дистрибутив системы управления сайтами на сервер. При помощи команды `wget` *линк дистрибутива*, грузим систему, распаковываем.

Важно подготовить базу данных с прикреплённым к ней пользователем, что можно сделать через интерфейс `phpMyAdmin` или программу, например, `Navicat`.

Далее собственно начинается то, ради чего используются системы управления сайтами. Система имеет алгоритм установки, в нашем случае достаточно набрать домен `school.lc` появляется следующее окно:

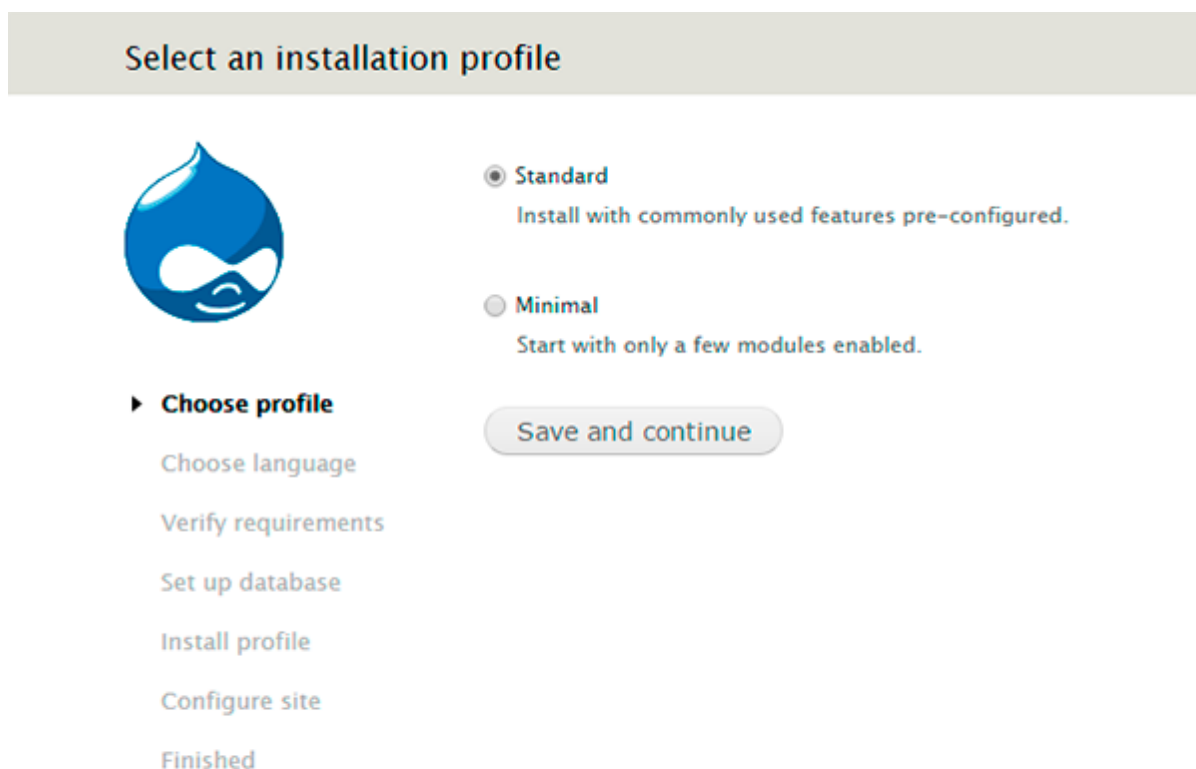


Рисунок 2 – Установка Drupal 7

Выбираем стандарт и дальше по шагам, где понадобится указать базу, логин и пароль для входа в административную панель управления.

1.5 Drupal 7 и её возможности

Согласно техническому заданию см. приложение А, нам понадобится строгое распределение пользователей по группам. У каждой группы должны быть свои привилегии. Основная возможность системы – это регистрация пользователей, их авторизация, возможность создания ролей помимо существующих: анонимный, зарегистрированный пользователь и суперпользователь. Распределение прав основано на ролях.

Права доступа позволяют управлять возможностями пользователей на сайте. Каждая роль имеет свой набор прав. Например, можно дать пользователям роли «зарегистрированный пользователь» право добавлять комментарии, а пользователям роли «анонимный посетитель» такого права не давать. Все права данные роли «зарегистрированный пользователь» распространяются на пользователей всех ролей, за исключением роли «анонимный посетитель». Для управления сайтом можно выбрать роль, которой будут автоматически назначаться все возможные права.

Права доступа позволяют управлять возможностями пользователей на сайте. Каждая **роль** имеет свой набор прав. Например, можно дать пользователям роли «зарегистрированный пользователь» право добавлять комментарии, а пользователям роли «анонимный посетитель» этого права не давать. Все права данной роли «зарегистрированный пользователь» распространяются на пользователей всех ролей, за исключением роли «анонимный посетитель». Для управления сайтом **можно выбрать** роль, которой будут автоматически назначаться все возможные права.

[Показать описание](#)

ПРАВО ДОСТУПА	АНОНИМНЫЙ ПОЛЬЗОВАТЕЛЬ	АВТОРИЗОВАННЫЙ ПОЛЬЗОВАТЕЛЬ	ADMINISTRATOR
Block			
Управление блоками	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Comment			
Управление комментариями и их настройками	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Просмотр комментариев	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Добавление комментариев	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Добавление комментариев без одобрения	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Правка своих комментариев	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Contextual links			
Использование контекстных ссылок	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Dashboard			
Просмотр административной панели управления	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Field			
Администрирование полей	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Filter			
Управление текстовыми форматами и фильтрами	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 3 – Таблица распределения прав по ролям

Примеры ролей: «анонимный посетитель», «зарегистрированный пользователь», «администратор», «модератор», «редактор» и т.д. Для удаления роли, выберите ссылку «правка», напротив её названия.

По умолчанию, Drupal поставляется с двумя пользовательскими ролями:

- анонимный пользователь – роль используется для пользователей, которые не имеют учётной записи или не авторизованы;
- авторизованный пользователь – роль автоматически применяется ко всем вошедшим под своим именем пользователям.

Другая основная особенность системы её расширяемость, т.е. функционал системы может быть дополнен за счёт устанавливаемых модулей. Модули устанавливаются в соответствии с потребностями, в случае отсутствия необходимого модуля можно произвести разработку собственного так называемого custom module в котором можно реализовать как востребованные hooks (крюки) для внесения изменения в систему без изменения ядра системы, так и собственный востребованный функционал.

Предполагается косвенное рассмотрение возможностей системы при реализации проекта, так как описание всех возможностей в быстро развивающемся проекте невозможно.

2 Реализация первой стадии разработки системы обучения

2.1 Структура системы

Имея на руках техническое задание какой-либо организации, написанное исходя из потребностей этой организации, остаётся техническая реализация исполнителя, что по сути вкладывается в определённые алгоритмы, в том числе существует даже научный подход решения этой проблемы.

Основа же лежит в структуре системы, на основе которой реализуются потребности. Касаясь исключительно Drupal и системы обучения остаётся предусмотреть минимальный набор необходимый для нормального функционирования обучающей системы:

- система, имея ядро должна обновляется независимо от модулей, исходя из соображений безопасности;
- должен быть единый стандарт API;
- дизайн и оформление необходимо вынести в отдельную категорию, сопряжённую с функционалом только по структуре распределения блоков сайта: как меню, слайдеры и т.п.

Реализация данной структуры происходит на стадии разработки древа сайта относительно корневой директории. Что имеется в CMS.

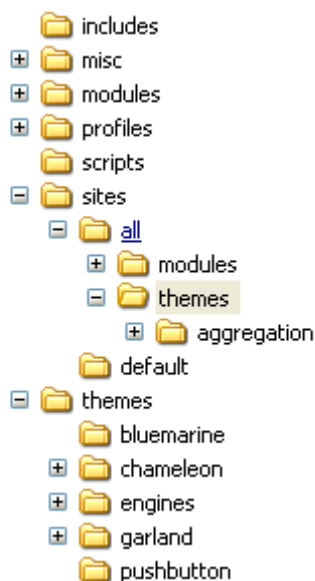


Рисунок 4 – Часть корневой структуры

Собственно, вынесение устанавливаемых модулей и тем оформления сайта со всеми файлами загруженными пользователями в отдельный каталог sites позволяет обновлять ядро, например, Drupal 7.23 → 7.34 при выявлении ошибок или проблем безопасности. Платформа по сути позволяет содержать несколько сайтов на разных доменах на одном файловом ядре и базе данных, что в целом не рекомендуется в силу возникновения путаницы и повышенной загруженности.

Это поверхностная структура позволяет не затрагивать рабочие файлы.

Обновление базы происходит за счёт встроенного скрипта `update.php` в корневой директории.

С технической точки зрения мы отделили понятия система реализации проекта и её тех. обслуживание от потребностей проекта и его функционирования, независимо, от системы реализации, которая не должна влиять на сам проект в отдельные промежутки времени.

2.2 Планирование проекта

Необходимо разобраться какие модули понадобятся для проекта. Стоит отметить, что отдельные модули, например, как Views представляют из себя сложный интерфейс с элементами программирования, аргументами (динамические фильтры), критерием которых обычно является значения, полученные из url. И реальные возможности отдельного модуля становятся ясны не сразу. В силу количества модулей, отдельный человек не может изучить их все. Например, модуль Rules позволяет реагировать на определенные события, добавлять условия и выполнять в зависимости от них действия.

В Rules оперируют 3-мя сущностями:

- событие (авторизация пользователя, удаление ноды и т.д.);
- условие (тип ноды, роль пользователя и т.д.);
- действие (создание ноды, обновление значения поля, редирект и т.д.).

По понятным причинам сопоставление потребностей проекта с тех. частью, что косвенно обычно относиться к его стоимости, может занять много времени, отсюда частые ошибки в оценках сроков и стоимости.

В нашем случае мы попытаемся разобраться с этим подробнее. Имея первую часть технического задания (ТЗ) приложение А, необходимо сопоставить потребности с имеющимися возможностями нашей системы. При регистрации пользователь должен быть определён как ученик, учитель или родитель ученика. Роли пользователей встроенный функционал и задать новые через интерфейс не составит сложности.

Выбор ролей при регистрации относиться к расширяемому функционалу, т.е. из нескольких похожих модулей нам нужно выбрать правильный. `Select registration roles` – по сути подходит для решения задачи, но по сравнению с другим не столь популярен. Выбрал `Auto Assign Role`.

Также у каждой роли есть свои поля данных указанных в ТЗ. Надо отметить, что я работал со многими CMS и часто вставлял банальный вопрос добавления полей в ту или иную форму. Если в системе нет простого способа сделать это, только программированием и обновлением базы данных не тратьте ваше время.

Drupal включает несколько типов документов по умолчанию – страницы, блоги, форумы, статьи и книги, но они достаточно просты и весьма ограничены. Модуль `Content Construction Kit (ССК)` позволяет создать свой собственный тип или добавить новые поля к основным типам содержания.

В нашем случае данный конструктор с поддержкой наиболее популярных типов полей как: число, текст, длинный текст и т.п. (рисунок 5) входит в ядро

системы и не является расширяемым функционалом.

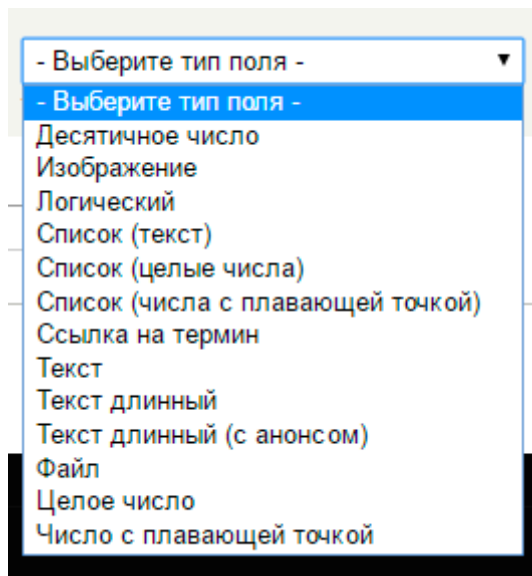


Рисунок 5 – Типы полей

Возвращаясь к пользователям, имеем вкладку управления полями и можем добавить поле телефон в форму регистрации, поле необходимо для всех ролей. Что в целом не решает нашей проблемы. Далее идут поля, относящиеся к той или иной роли, только в отдельности.

Необходимы профайлы, установив Profile 2 у пользователей появляется новая вкладка где он может указать данные относящиеся к его роли. Модуль позволяет создать отдельные страницы регистрации для каждого профайла, но он не интегрирован с Auto Assign Role и перенаправления на нужную страницу после выбора роли при регистрации, а ещё лучше, как это обычно просят, загрузки формы без перезагрузки страницы при выборе роли не происходит.

Собственно, тут и становятся основные задачи планирования, опытные специалисты не станут тратить время на и так работающий, возможно не в той форме функционал. Оценка времени же и средств по интеграции модулей, уходит в тех. часть, API и написание модулей. И берётся в среднем исходят из строк в час специалиста проработавшего полгода с системой и приблизительное количество необходимых строк кода для решения проблемы. Считается по всей длине проекта. Из-за этого планирование может занять до 80% времени выполнения проекта.

Сейчас никто этим серьёзно не занимается и хорошо, что есть готовые системы управления сайтами где можно, порой очень сложно, найти компромисс.

В нашем случае, исходя из ситуации, мы оставим заполнение профайлов пользователями, на часть сценария, нашей системы, относящуюся к после регистрационному периоду.

2.3 Техническая часть

В технической части можно остановиться на программной составляющей системы. Начать можно с того, что система выстроена не на объектное ориентированном подходе, а на крюках.

Хук – это «крюк», обрабатывается ядром Drupal и служит элементом для построения модуля. Каждый хук идет как событие. К примеру, `hook_user_insert` вызывается тогда, когда в системе создается новый пользователь. То есть у нас есть событие: создание пользователя, `hook_init` запускается всегда перед началом инициализации страницы, а вот `hook_menu` запускается каждый раз при очистке кеша и с его помощью можно внедрить в Drupal свои страницы. Мы остановимся подробнее позже на `hook_menu`, так как написание модуля обычно начинается с этого крюка.

Сами разработчики пытались объяснить сей подход отсутствием поддержки объектно ориентированного программирования (ООП) на ранних стадиях разработки системы у РНР. По-моему, получилось даже лучше. Есть описание всех крюков и событий в системе, которые они вызывают при инициализации в том или ином модуле.

Обработка же запросов, поступающих в систему, выполнена следующим образом. Имея стандартные настройки Apache2 загружает файл `index.php` из корневой директории. В индексе четыре строки:

1. `define('DRUPAL_ROOT', getcwd());`
2. `require_once DRUPAL_ROOT . '/includes/bootstrap.inc';`
3. `drupal_bootstrap(DRUPAL_BOOTSTRAP_FULL);`
4. `menu_execute_active_handler();`

Первая строка определяет путь в установочную директорию. Вторая загружает РНР код в файле `bootstrap.inc`. Расширения `inc` используется для защиты от стороннего запуска файла его прямым указанием в адресной строке. В третьей строке происходит инициализация системы, проверка соединения с базой, загрузка модулей, загрузка данных в память. Четвёртая строка обрабатывает запросы, поступающие из адресной строки браузера `http://domain.com/?q=node/234` Нас интересует часть `?q=node/234` где система определяет является ли это запрос к узлу, таксономии, профайлу или динамической странице реализованной в Видах.

Bootstrap процесс имеет 8 фаз инициализации:

```
/**
 * First bootstrap phase: initialize configuration.
 */
define('DRUPAL_BOOTSTRAP_CONFIGURATION', 0);
/**
 * Second bootstrap phase: try to serve a cached page.
 */
define('DRUPAL_BOOTSTRAP_PAGE_CACHE', 1);
/**
```

```

* Third bootstrap phase: initialize database layer.
*/
define('DRUPAL_BOOTSTRAP_DATABASE', 2);
/**
* Fourth bootstrap phase: initialize the variable system.
*/
define('DRUPAL_BOOTSTRAP_VARIABLES', 3);
/**
* Fifth bootstrap phase: initialize session handling.
*/
define('DRUPAL_BOOTSTRAP_SESSION', 4);
/**
* Sixth bootstrap phase: set up the page header.
*/
define('DRUPAL_BOOTSTRAP_PAGE_HEADER', 5);
/**
* Seventh bootstrap phase: find out language of the page.
*/
define('DRUPAL_BOOTSTRAP_LANGUAGE', 6);
/**
* Final bootstrap phase: Drupal is fully loaded; validate and fix input data.
*/
define('DRUPAL_BOOTSTRAP_FULL', 7);

```

DRUPAL_BOOTSTRAP_CONFIGURATION – устанавливает обработчики ошибок и исключений, работает с `$_SERVER` массивом, устанавливает конфигурацию для сообщения об ошибках, устанавливает переменные сеанса, используя `ini_set()`. Процесс включает в себя загрузку файла `settings.php`, который содержит настройки подключения к базе данных и другие вещи. Создает много разных глобальных переменных, таких как `$cookie_domain`, `$conf`, `$is_https` и другое.

DRUPAL_BOOTSTRAP_PAGE_CACHE – загрузка страницы идет из кеша. Проверяет занесен ли пользователь в черный список, и если это так возвращает 403 ошибку. Если пользователь имеет валидный сессионной cookie, то просмотр авторизованным пользователям сохранённой страницы система запрещает. Также собирается глобальный объект `$user`.

DRUPAL_BOOTSTRAP_DATABASE – если не установлена переменная `$GLOBALS ['databases']` и не начат процесс установки, тогда мы загружаем `/install.php`, так как Drupal предполагает, что мы должны установить сайт. Подключается файл `database.inc`, который загружает все функции абстракции базы данных которые мы знаем и любим, такие как `db_query()` и `db_select()` и `db_update()`. Также он содержит базовые классы `Database` и `DatabaseConnection` и `DatabaseTransaction` (среди кучи других). Это файл размером в 3000+ строк, это выходит за рамки обсуждения первоначальной загрузки.

DRUPAL_BOOTSTRAP_VARIABLES – загружает переменные из таблицы.

Переменные сохранённые в таблице, во время инициализации проверяются в settings.php, module.inc и любом .module файле, затем в таблице cache_bootstrap и толь в конце из оригинального места variable таблицы базы данных.

DRUPAL_BOOTSTRAP_SESSION – заполняет собранный ранее объект \$user подгружая данные из базы. Работает только для авторизованного пользователя.

DRUPAL_BOOTSTRAP_PAGE_HEADER – отправляет http заголовки страницы.

Content-Type: text/html;charset=windows-1251

Allow: GET,HEAD,OPTIONS

Content-Length: 1984

И тому подобное.

DRUPAL_BOOTSTRAP_LANGUAGE – если сайт многоязычный, инициализируется тип языка.

DRUPAL_BOOTSTRAP_FULL – последняя стадия инициализации подгружаются файлы: ajax.inc, mail.inc, token.inc, image.inc, file.inc. Самые тяжёлый и ресурсоёмки функционал в стадии инициализации. Также загружаются все включенные модули.

Возвращаясь снова к программной части и крюкам рассмотрим hook_menu. По сути при определении модуля, для создания которого в директории модулей папки sites нужно создать саму папку модуля и файл .info содержащий минимальный набор данных:

name = Читабельное название модуля

description = Одно-два предложения, описывающие назначение модуля

core = 7.x

Также указываем файл .module, который может содержать hook_menu(). Сам файл является телом модуля и подгружается автоматически. Название же хука можно разложить на две составляющих, первая часть hook меняется на машинное имя модуля, вторая часть menu по сути является служебным словом и не может быть заменено или использоваться где-либо ещё. Каждый хук имеет свой уникальный синтаксис, набор командных слов, словосочетаний.

В нашем случае предназначение хука определено как регистрация URL в системе для вызова того или иного функционала модуля. У нас уже есть административная панель управления с разными разделами, в которых указаны ссылки на изменение как контента, так и таксономии со словарями и т.п. Наш хук интегрирует путь доступа к написанным нами функциям. Происходит это следующим образом:

```
function mymodule_menu() {
    $items['admin/config/system/foo'] = array(
        'title' => 'Foo settings',
        'type' => MENU_NORMAL_ITEM,
        // Другие элементы должны быть добавлены здесь.
    );
    return $items;
}
```

admin/config/system – системный путь к которому мы добавили foo название модуля в url несопряжённое с реальным названием. Title – как мы видим заголовок в разделе система панели управления. Type – обычная регистрация URL может управляться администратором и идёт по умолчанию указывать необязательно. Другие элементы, без которых не работает наш вызов функции модуля:

```
'access arguments' => array('administer autoassignrole'),  
'page callback' => 'autoassignrole_admin_block_page',  
'file' => 'autoassignrole.admin.inc',
```

Access arguments – управление доступом, кто может перейти по ссылке. Page callback – вызов функции, обрабатывающей запрос. File – файл в котором находится функция в пределах директории модуля. Примерно так начинается написание модуля. При этом название крюки полностью соответствует действительности, система сама загружает модуль находя его в директории где его ожидает, также распознаёт системные файлы info и module, далее парсит файлы находит системное описание модуля и крюки.

Ещё стоит обратить внимание на файл .install, который можно разместить в директории модуля, тоже является частью системы и выполняется при включение модуля. Ограничен по используемым в нём хукам, по сути служит для создания таблиц и полей в базе с которыми работает модуль, также может обновлять поля и таблицы при обновлении модуля, производит удаление своих таблиц при выборе опции в административной панели управления.

При правильном использовании возможностей системы время написания модуля сильно сокращается, так как мы оперируем уже готовыми составляющими. Также возможно использовать функционал других модулей и обращаться к ним через их API часто тоже реализованный на крюках. Как например в модуле Views существует файл views.api.php и достаточно много поддерживаемых им крюков.

2.4 Интерфейс, панель управления обучающей системы

Интерфейс администратора можно отнести к программируемому интерфейсу как логически, используются Токены – замещают составляющую системы в виде определённого символьного шаблона. Например, есть юзер ID и нам необходимо вывести его в URL, задать в правилах или указать в фильтрах, мы используем системный токен [user-id] в интерфейсе в необходимом разделе. На самом деле интерфейс позволяет использовать языки программирования как PHP, так и их современные производные PHPQuery и некоторые другие.

Для этого в модулях, раздел ядро, необходимо активировать PHP filter. Ярким примером возможностей интерфейса может служить Views модуль, глобальное пользовательское поле с tokens – шаблонами передзагруженных полей в предыдущем окне иерархической цепочке интерфейса Видов рисунок 6.

Текст

Введите текст, отображаемый полем. Вы можете использовать HTML, а также подстановочные шаблоны, указанные ниже.

ПОДСТАНОВОЧНЫЕ ШАБЛОНЫ

Для данного поля доступны следующие маркеры. Обратите внимание, что из-за порядка обработки, невозможно использовать поля, находящиеся ниже данного поля; если нужное поле отсутствует в списке, переставьте поля. Если необходимо вставить символы " и ', используйте их html коды """ или "'", иначе вместо них будут пробелы.

<Поля>

- [title] == Содержимое: Заголовок
- [uid] == Содержимое: UID Автора
- [changed] == Содержимое: Дата обновления
- [nothing] == Глобальный: Пользовательский текст

Рисунок 6 – Глобальное пользовательское поле

Ещё более сложный модуль Rules, который может взаимодействовать с модулем Token расширяющим линейку шаблонов составляющих частей системы. В большей степени это следствие сложной первичной структуры самого интерфейса и его сущностей: типов контента, таксономии со словарями, термов, меню, пользователей, ролей и т.п. с чем мы и попытаемся ознакомиться.

Сущности (entities) – отличный способ организации данных в Drupal. Узлы (node), термы в таксономии, комментарии или пользователи в Drupal, все эти вещи, начиная с 7 версии Drupal, являются сущностями.

Рассмотрим основные составляющие нашего интерфейса:

– Node – узел системы, тип контента, состоит из заголовка, тела и других полей добавленных администратором. Фактически является страницей, картой товара и т.п.;

– Таксономия – структура узлов, их словари и термы, логическая цепочка;

– Блоки – фактическое расположение в рабочем пространстве сайта того или иного элемента;

– Меню – указатель путей для страниц, функций сайта.

Понятно, что составляющие системы могут расширяться, с введением сущностей в 7-мой версии Друпал написание составляющих стало возможным.

Изучение всех возможностей системы упирается в бесконечно нарастающей в ширь и глубину (сложность) функционал системы. И не является возможным для индивида. Поэтому перейдём к конкретному случаю.

3 Реализация проекта

3.1 Расписание занятий

Для создания расписания по ТЗ приложения А нам понадобится модуль Event Calendar имеющий интеграцию с Views. Модификации подлежит представление event-created/week. Основная задача определиться с файлом оформления расписания, в нашем случае он вынесен в модуле Calendar в директории theme и называется calendar-week-overlap.tpl.php. Все типовые файлы *.tpl.php нужно переписывать в текущем шаблоне (дизайн сайта).

Следовательно, необходимо создать тему нашего сайта. Позаимствуем тему с сайта drupal.org с названием Education. Следуя структуре темы, переписываем файл calendar-week-overlap.tpl.php сохраняя его в директорию sites/all/themes/education/theme/tpl. Директории могут иметь любую структуру и названия, файлы *.tpl.php система находит автоматически. Поэтому обычно исходят из логики обычного удобства, т.е. если проект большой, то файлы для представлений в своей папке и всё остальное тоже разложено по своим папкам.

Само программирование под Drupal сложно даже на первичных стадиях, а без дебаггеров невозможно. Сами дебаггеры развиты слабо. Ставим FirePHP 4, для этого в FireFox нужен FireBug. После установки всех дополнительных расширений в браузере, подключаем FirePHP в Drupal в файле index.php

```
require 'FirePHPCore/fb.php';
```

При этом в корневую директорию нужно загрузить FirePHPCore. Получаем развёртку массива или объекта при указании

```
fb($переменная);
```

Пример на рисунке 7 *fb(\$cell)*; в файле calendar-week-overlap.tpl.php

```
http://school.lc/event-created/week#overlay-context=
array('header'=>'1', 'class'=>'days mon', 'data'=> ... )
array('header'=>'1', 'class'=>'days tue', 'data'=> ... )
array('header'=>'1', 'class'=>'days wed', 'data'=> ... )
array('header'=>'1', 'class'=>'days thu', 'data'=> ... )
array('header'=>'1', 'class'=>'days fri', 'data'=> ... )
array('header'=>'1', 'class'=>'days sat', 'data'=> ... )
array('header'=>'1', 'class'=>'days sun', 'data'=> ... )
```



Рисунок 7 – FirePHP дебаггер

В системе присутствуют очень большие массивы и объекты при этом производить выборку данных при программировании в интерфейсе ещё проблематичней и порой нужны другие решения.

Для создания расписания нам придётся изменить логику работы календаря-событий. У нас есть таблица часов от 0 до 24 по вертикали и дни недели по горизонтали. Нужно сделать дни недели по вертикали с часами и название классов по горизонтали.

Файл оформления представления позволяет нам это сделать, задав массив для 5-х классов вида

```
$mass = array(1 => 'А', 2 => 'Б', 3 => 'В', 4 => 'Г', 5 => 'Д');
```

мы раскладываем дни недели не в столбик, а в линейку, изменяя генерацию разложения массива с днями недели. В общем собственно прибегая к обычному нехитрому программированию добиваемся положительного эффекта см. рисунок 8.

	Класс 5 А	Класс 5 Б	Класс 5 В	Класс 5 Г	Класс 5 Д
Пятница					
Перед 01					
01					
02					
03					
04					
05					
06					
07					
08					
09				Химия ЕОТ	Математика ЕОТ
10				Математика ЕОТ	
11					

Рисунок 8 – Расписание

Часть файла:

```
/**
 * Далее происходит фильтрация ноды для
 * расположения в сетке расписания **/
If
(!empty($items[$start_time]['values'][0]))
    { $day_of_w = $mass_days[0]; }
elseif
(!empty($items[$start_time]['values'][$index]))
    {$day_of_w = $mass_days[$index];}
    if (isset($day_of_w)) {
        if ($cell['data'] == $day_of_w) { ?>
<?php for ($l = 0; $l <= 5; $l++) {
If (!empty($items[$start_time]['values'][$l][0]['item']-
>rendered_fields['field_namecl'])) {
if ($items[$start_time]['values'][$l][0]['item']->rendered_fields['field_namecl']
== $mass[$index] ) { ?>
<?php foreach($items[$start_time]['values'][$l] as $item) : ?>
<?php if (isset($item['is_first']) && $item['is_first']) :?>
<div class="item <?php print $item['class']?> first_item">
<?php else : ?>
<div class="item <?php print $item['class']?>">
<?php endif; ?>
<?php print $item['entry']; // вывод заголовка ноды
```

```
// в расписание(название предмета)?>
</div>
<?php endforeach; ?>
<?php }}} ?>
<?php }}
/*****/
?>
```

3.2 Оформление проекта

Стоит остановиться на оформлении проекта, для этого достаточно время затратного процесса придуман специальный подход. Когда-то давно в информационных системах (ИС) не делили логику (функционал) и темизацию проекта. Писав всё вместе, в корни неверный подход, хотя и требует много больше времени для его решения.

Тема оформления – это набор файлов, отвечающих за внешний вид сайта. Тема оформления определяет HTML-макет страницы, используемые на ней CSS-файлы и JavaScript. Несмотря на то что шаблоны, используемые в темах оформления, позволяют исполнять PHP-код, в них не должно быть кода, не связанного с отображением информации. Тема оформления должна на входе получать готовые массивы данных, а на выходе создавать из них HTML-код. Логика программы реализуется в модулях.

Все страницы разделяются на регионы – области страницы, каждая из которых может содержать один или несколько блоков с данными. Например, по умолчанию большинство стандартных тем оформления состоит из пяти регионов: левая и правая колонка, шапка, подвал сайта и регион с основным его содержимым. Такой макет страницы наиболее популярен, поскольку удовлетворяет требованиям большинства пользователей; с его помощью удобно строить новостные сайты и блоги. Разработчик может определять в макете страницы любое количество дополнительных регионов и размещать их в любых самых неожиданных местах. Задание регионов происходит в файле *.info и имеет вид

```
regions[page_top] = Page top
Обычный листинг *.info файла
name = Theme name
description = Theme description.
version = VERSION
core = 7.x
engine = phptemplate
stylesheets[all][] = style.css
stylesheets[print][] = print.css
scripts[] = effects.js
scripts[] = menu.js
regions[left] = Left sidebar
```

```
regions[content] = Content
regions[header] = Header
regions[footer] = Footer
```

Блок – это набор особым образом сформатированных данных. Внешний вид и содержимое блока определяются создающим его модулем и функциями темизации, а расположение – администратором сайта. Используя специальный графический интерфейс, администратор может методом буксировки (drag-and-drop) разместить любой блок системы в любом из доступных регионов. Примером блока может служить форма входа в систему или облако тегов.

RNPTemplate при сборке страницы берет информацию из пяти основных файлов: имя_темы.info, page.tpl.php, node.tpl.php, block.tpl.php, template.php. Если же включен модуль comment, для вывода комментариев используется шаблон comment.tpl.php. Собственно RNPTemplate это своего рода подсистема обрабатывающая шаблон сайта.

Page.tpl.php – это основной шаблон, в нем размещается HTML-код всей страницы, которая позже будет выведена пользователю. В частях страницы, где будет выводиться содержимое регионов и основной текст, должны располагаться соответствующие переменные. Например, если в *.info-файле не указано иное, то regions[page_top] = Page top соответствует переменная массива \$page['page_top']. И обычно задаётся

```
if($page_top= render($page['page_top']))
то
print $page_top
```

В регионе может размещаться что угодно, настроенное в интерфейсе или же созданное в модулях или же при совместном взаимодействии. Т.е. обычно своего рода подготовленный HTML код.

Что стоит помнить, оформляя проект, при современном разнообразии браузеров и операционных систем их версий и тому подобное. Все JavaScript-ы обрабатываются на стороне пользователя везде по-разному, один и тот же браузер на разных ОС работает по-разному. И из-за этого в корректном выводе информации всегда приходится сомневаться. Нет даже единой базы проверки шаблонов, а устанавливать все ОС и браузеры проблематично. Ещё сложнее при подключение всего разнообразия современных устройств.

3.3 Использование расписания

Создав расписание из модулей: Календарь (calendar), Календарь событий (event calendar) и Представлений (views) необходимо его правильно использовать. Т.е. есть ответственные за составление расписания и у них должен быть определённый доступ. В системе распределение прав происходит по ролям, было создано три роли: Ученик, Преподаватель, Родитель. Также по умолчанию имеется суперпользователь. Ни одной из ролей мы не можем доверить редактирование общедоступного расписания, необходимо создать роль Модератор.

Далее стоит отметить особенность распределения прав доступа она

происходит в зависимости от события. Не просто выделением типа материала для управления модератором, а по действиям, которые можно совершить с содержимым, что очень удобно:

- Предметы: Создание новых материалов;
- Предметы: Редактирование собственных материалов;
- Предметы: Редактирование любых материалов;
- Предметы: Удаление собственных материалов;
- Предметы: Удаление любых материалов.

Указав линк на представление расписания в меню сайта делаем его общедоступным для пользователей. Теперь любой желающий может узнать время проведения предмета у любого класса.

3.4 Создание дневника

На первый взгляд создание дневника, имея расписание, что по сути подобно дневнику за неимением дополнительных полей задание и оценка легко. На деле же дневник должен быть привязан к определённому ученику и доступ к нему должен быть ограничен, но в тоже время удобен. Попытаемся с этим разобраться.

Начать можно с создания принципиальной страницы дневника. Для этого клонируем страницу расписания в Представлениях. Клонирование выбрано неспроста, наследует все предыдущие настройки и функции. Изменяем название вкладки, машинное имя. При определённом наборе полей в типе материала для расписания сохраняем те же поля в новом типе материала для дневника. Изменяем критерий фильтрации в настройках представления:

Содержимое: Тип (= Дневник)

Для редактирования шаблона представления предусмотрена спинальная система распознавания имён файлов для самого представления так и его частей рисунок 9.

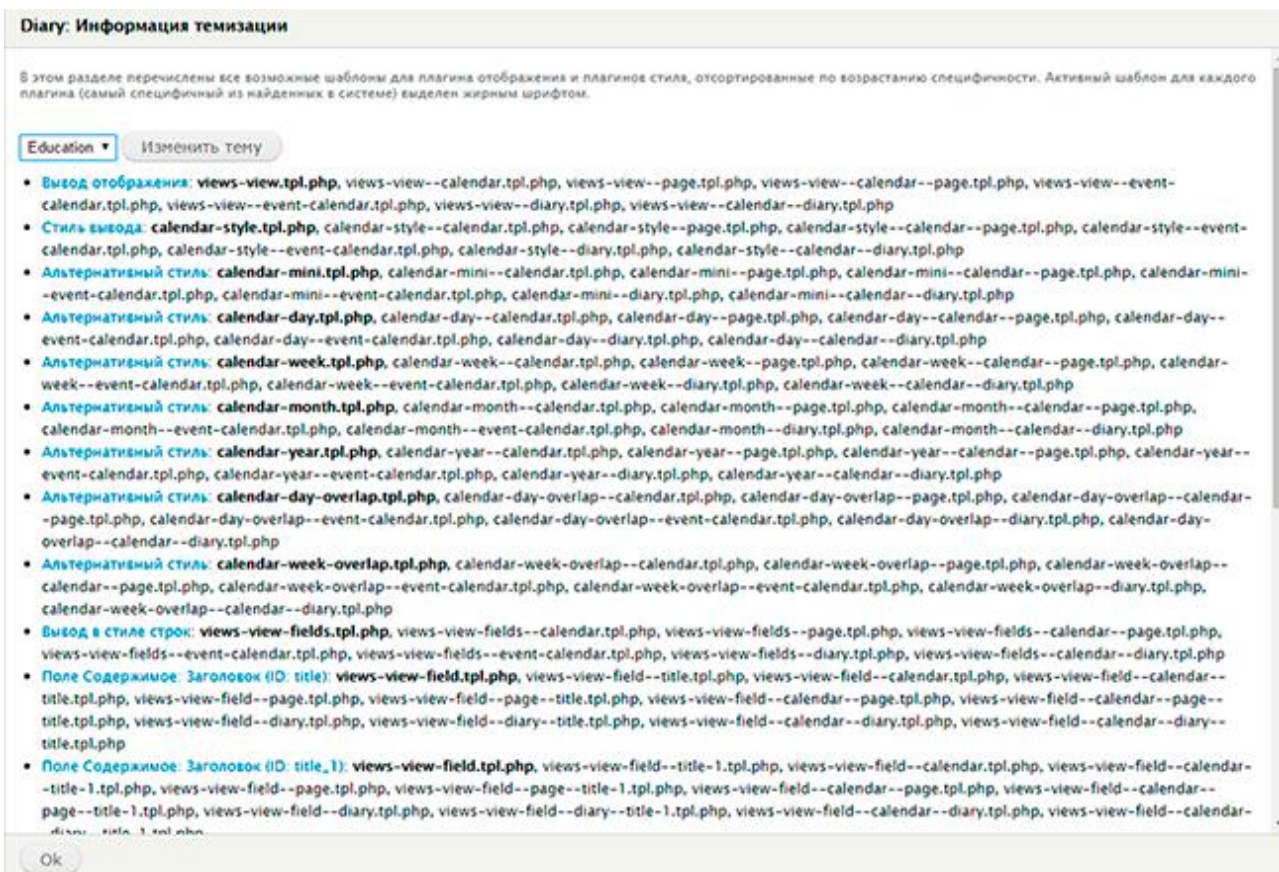


Рисунок 9 – Распознавания имён файлов шаблонов представления

Переопределив файл шаблона для представления и добавив необходимые поля задание и оценка в тип материала «Дневник» создаём страницу дневника.

3.5 Создание раздела с тестами

Раздел тестирования носит сугубо технический характер. Так как его применение в школе на сегодняшний день не определено. Он может быть использован для определения способностей в направлении обучения, гуманитарные или технические классы. Так и для тестирования по определённым предметам в школе или же дома в качестве домашнего задания.

Если рассматривать данный раздел в этом ключе, то его создание достаточно сложное начинание, в плане гибкости настраиваемых типов контента и возможностей для даваемых ответов на вопросы. Формулу ввести достаточно проблематично так же, как и тригонометрические значения.

Для создания на сайте данного раздела воспользуемся модулем Quiz имеющим интеграцию с Views и Rules через модуль сущностей (Entity API). Самое интересное в нём интеграция с H5P модулем, как ранее упоминалось необходима гибкость в создание контента. По умолчанию данный модуль вносит 20 типов контента:

- наборы вопросов;
- вопросы с возможностью перетаскивания (drag and drop) правильного ответа;

- многозадачные вопросы, с возможностью давать несколько ответов;
- настольные игры;
- флешь карточки.

Так как это типы материалов, то использовать их можно по-разному. Сам модуль Quiz имеет встроенный гибкий таймер для тестов. Перетасовку вопросов для теста, при ограниченном выводе, например, в 20 вопросов, вывод перетасовывается имея в наличие большую базу. Имеет интеграцию с HTML5.

Для активации модуля понадобится установить следующие составляющие проекта, отсутствующие в директории модулей:

- Rules;
- Token;
- Views_bulk_operations.

Надо заметить, что сопутствующие модули необходимые для нормального функционирования устанавливаемого, указываются в файле *.info в виде зависимостей:

```
dependencies[] = ctools
dependencies[] = entity
dependencies[] = filter
dependencies[] = rules
dependencies[] = token
dependencies[] = views
dependencies[] = views_bulk_operations
```

Загрузка и установка модулей происходит стандартным образом, описанным ранее в разделе 1.4. Сам модуль тестирования разделён на три части, Quiz core (ядро), Quiz question (типы тестов), Quiz addon (дополнения).

Вместе с активацией основного модуля из ядра происходит активация сопутствующих модулей, которые предварительно были загружены. Модуль Quiz настолько большой, что создаёт в системе целый раздел на уровне других разделов как: конфигурация, модули и т.п. рисунок 10.

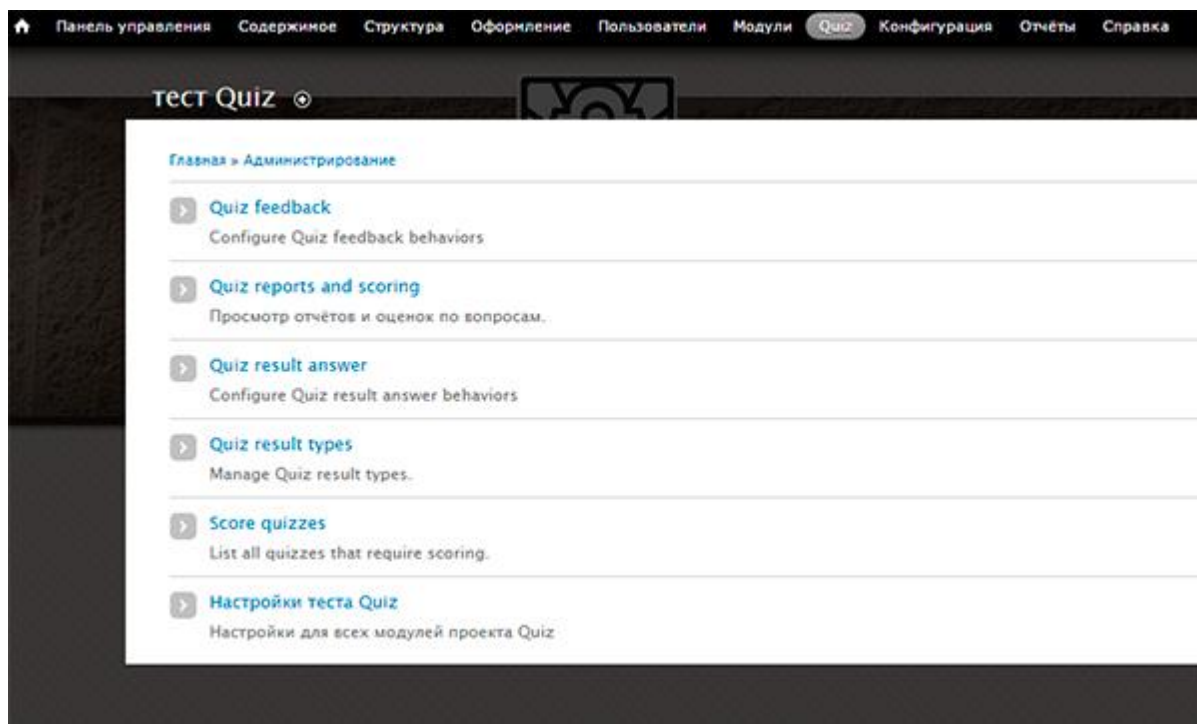


Рисунок 10 – Раздел управления тестами

В разделе присутствуют такие сопутствующие подразделы, как отчёты и оценка (reports and scoring), результаты/ответы, настройка тестов, управление результатами, список тестов нуждающихся в оценке. При всём этом активирована была только основная часть модуля. Ядро содержит ещё две основные части как: Quiz – Directions и Quiz – Pages.

Далее идут вспомогательные модули, располагающиеся в древе файлов в директории quiz/modules их мы видим в панели управления как Quiz addon. В директории quiz/question_types находятся подмодули Quiz question. Каждый подмодуль имеет свой *.info файл, сообщающий системе, что по сути это индивидуальная структурная единица.

Остаётся только верно задать алгоритм работы тестов, время теста, его оценивание и т.д.

4 Дополнительные сведения по конфигурации и защите проекта

4.1 Настройка FTP соединения, владельца и группы для файлов

Для работы с проектом на удалённой машине, где бы она не находилась, необходимо настроить FTP соединение, что расшифровывается как протокол передачи файлов. В этих целях используются специальные сервисы и программы как на сервере, так и на персональном компьютере.

На виртуальной машине нашего проекта установлен сервис *pure-ftp*. Его установка несложна и происходит стандартным образом *apt-get install pure-ftpd*. Особенности состоят в настройке сервиса, для доступа к директории сайта по логину и паролю через протокол FTP необходимо задать владельца и группу для файлов сайта отличную от суперпользователя. Воспользовавшись утилитой *unix useradd* генерируем следующую команду

```
useradd -d /var/www/school -s /dev/null ftpadmin > /dev/null 2>&1
```

Где флаг *d* – присваивает новому пользователю домашнюю директорию. А флаг *s* – указывает шелл для пользователя, которым он может пользоваться для входа на сервер. Так как у пользователя для FTP не должно быть шелл доступа прописываем ему нулевое значение, вывод самой команды тоже отправляем в нулевое значение.

Перенаправление вывода оператором «>», как в примере выше, переадресовывает вывод программы. В данном случае, что-то отправляется в */dev/null*, а что-то переадресовывается в *&1*. Существует три стандартных значения вывода для программ. Ввод получают от клавиатуры или из программы, получается обработка вывода другой программы. Результат программы обычно печатается в стандартный вывод и иногда в файл «STDERR» (ошибка). Все это три дескриптора, которые часто называют STDIN, STDOUT и STDERR. Но часто к ним обращаются не по имени, а по номеру: 0 – STDIN, 1 – STDOUT и 2 – STDERR. По умолчанию, если не указан номер, то будет подразумеваться STDOUT. В нашем примере видно, что команда направляет свой стандартный вывод в */dev/null* псевдоустройство, которое может принять произвольный объём данных. Затем все ошибки, то есть STDERR перенаправляет в стандартный вывод. Необходимо поставить амперсанд «&» перед номером назначения. Это один из способов сделать программу безмолвной. Добавлю, что команда в примере аналогична команде *command >/dev/null 2>/dev/null*.

Далее необходимо прописать */dev/null* в списке доступных шеллов */etc/shells*, что при попытке войти через шелл перенаправит пользователя на псевдоустройство. Задаём пароль пользователя FTP, что выполняется командой

```
passwd ftpadmin
```

Необходимо также перейти в директорию сайта и установить группу и владельца на те файлы, которые мы хотим разрешить редактировать этому пользователю через FTP, выполняется следующим образом

```
chown -R ftpadmin:ftpadmin директория
```

Где флаг *R* – рекурсивная обработка, т.е. можно указывать директории с

файлами, а не файлы в отдельности.

Настраиваем FTP клиент, тип соединения пассивный FTP. Для работы по FTP-протоколу используется два соединения: первое соединение для команд, а второе для данных. Соединение для команд устанавливает клиент, а для данных соответственно сервер. Но в некоторых случаях необходимо, чтобы оба соединения устанавливал клиент. В этом случае сервер переводится в «пассивный режим» командой «PASV». Пассивный режим необходимо использовать, если компьютер не имеет внешнего IP-адреса, например, если находится в локальной сети.

Теперь мы можем редактировать файлы сайта. Нужно помнить, что ИС часто пишет файлы самостоятельно. У Друпал есть специальная директория для всех загружаемых файлов, /sites/default/files. Изначально система просит разрешить запись в этой директории, есть два способа это сделать. Утилита `chmod` – программа для изменения прав доступа, принимает значения таблица 1.

Таблица 1 – Принимаемые значения

Двоичная	Восьмеричная	Символьная	Права на файл	Права на директорию
000	0	---	нет	нет
001	1	--x	выполнение	чтение файлов и их свойств
010	2	-w-	запись	нет
011	3	-wx	запись и выполнение	всё, кроме чтения списка файлов
100	4	r--	чтение	чтение имён файлов
101	5	r-x	чтение и выполнение	доступ на чтение
110	6	rw-	чтение и запись	чтение имён файлов
111	7	rwx	все права	все права

Например, `chmod 4555 {имяфайла}` – все имеют право на чтение и выполнение, но запускаться файл на исполнение будет с правами владельца. При присвоении прав 777 директории файлов система получит возможность записывать файлы и создавать папки, однако доступ к директории получают все владельцы и группы в UNIX. Для того что бы разрешить системе писать в директорию, а работает она под владельцем программы Apache2, рекомендуется директории files присвоить значение `www-data`, группа и владелец могут различаться, в нашем случае оба названия совпадают. Также устанавливаются права на изменение файлов, а система при установке требует разрешить на запись файл `settings.php` находящийся там же где и папка files. После установки ограниченные права доступа файлу можно вернуть.

Одна из особенностей FTP доступа, для отображения скрытых файлов в фильтре необходимо задать флаг «-a», что позволяет видеть файлы начинающиеся с точки. Для редактирования и установки модулей часто требуется файл `.htaccess`.

Программы клиенты, позволяющие редактировать и изменять файлы

автоматически, более удобны, нежели ручное перенесение изменённых файлов.

4.2 Настройка PHP языка программирования

Выбрав LAMP при установке Ubuntu устанавливается по умолчанию настроенный PHP язык программирования рисунок 11.

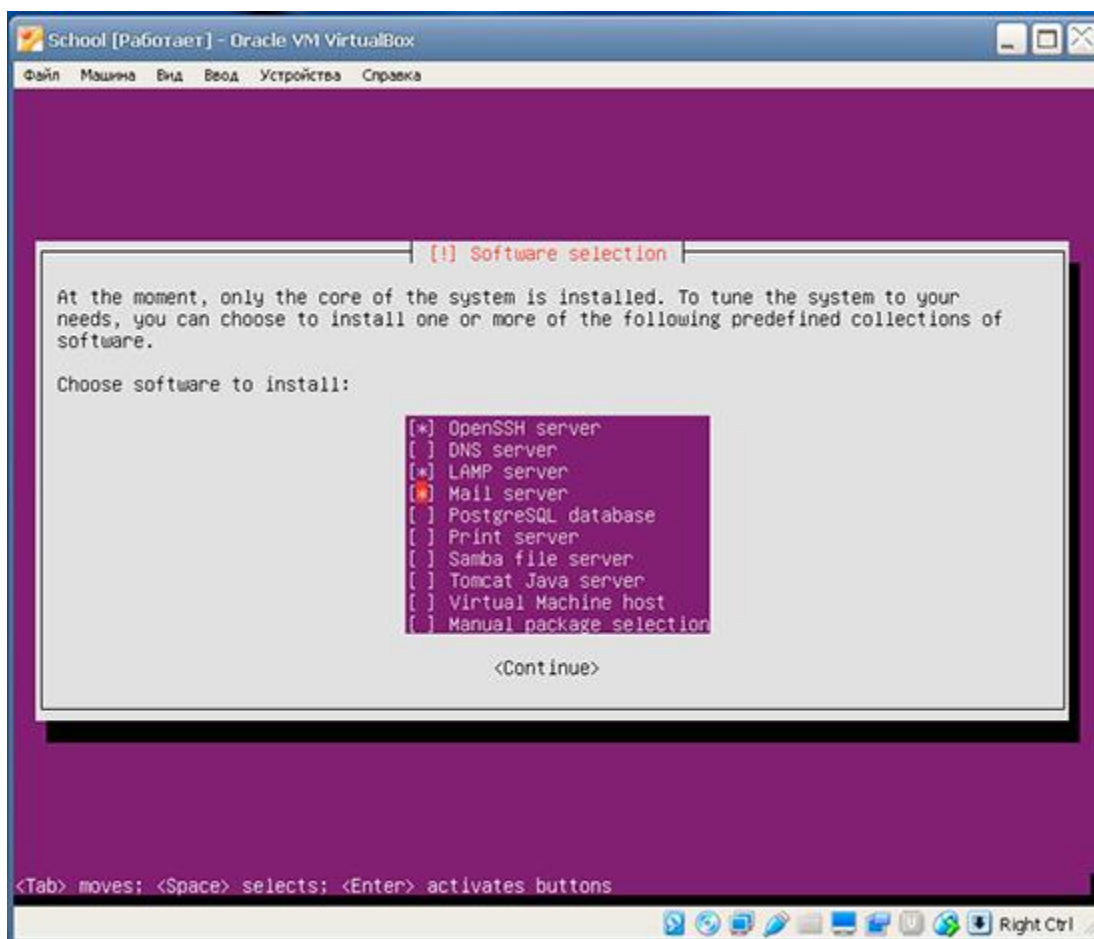


Рисунок 11 – Окно выбора необходимых сервисов во время установки ОС Ubuntu

Данный вариант не подходит для реализации проекта и PHP необходимо настроить. Файл для настройки PHP находится по адресу `/etc/php5/apache2/php.ini`. Открываем файл командой `riso -w` и находим строчку `memory_limit`, выставляем значение в мегабайтах. Если в планах есть использовать какие-либо инструменты для работы с изображениями и превью, нужно, как минимум, 128 МБ, так как изменение размера одного изображения уже потребует до 46 МБ. Рекомендуемый лимит 256 МБ. Следующий важный параметр для изменения `max_input_time` по умолчанию 60 секунд. Эта директива задает максимальное время в секундах, в течение которого скрипт должен переработать все входные данные, переданные запросами вроде POST или GET через форму данных. Это время измеряется от момента, когда PHP вызван на сервере до момента, когда скрипт начинает выполняться. Значение -1, означает,

что будет использоваться `max_execution_time`. Если установить равным 0, то ограничений по времени не будет. Если на сайт загружаются большие файлы, при этом порой необходима конвертация форматов, лимит нужно увеличить значительно, например, до 1 800 секунд, что равняется 30 минутам, так как скрипт сохранения файла сработает не сразу, а после подготовки файла.

Значение следующего параметра, сложно объединить с предыдущим, `max_execution_time` – максимальное время исполнения скрипта системы в отдельности, измеряется в секундах. Не рекомендуется ставить бездумно большим, из-за угроз безопасности: как вредоносная загрузка ресурсов и попыток взломов системы.

Есть ещё несколько параметров без которых успешное функционирование ИС невозможно, `post_max_size` устанавливает максимально допустимый размер данных, отправляемых методом POST. Это значение также влияет на загрузку файлов, через обычную форму. Для загрузки больших файлов это значение должно быть больше значения директивы `upload_max_filesize`. Так как форма может содержать другие данные. По умолчанию выставляются в мегабайтах можно перевести в байты.

Далее необходимо заблокировать использование опасных функций `disable_functions` динамическая директива и зависти от версии языка и установленных библиотек. Существуют PHP шеллы, которые ведут себя идентичное командной строке сервера, часто сервера остаются с открытым анонимным доступом, что по сути уже проникновение. Выполнять шелл команды позволяет не одна функция в PHP, также много других потенциально опасных функций.

При установке пакета LAMP PHP устанавливается с наиболее популярными библиотеками, но библиотека GD – графический редактор, не входит в это список её нужно установить, дополнительной конфигурации не требует.

Это основные настройки необходимые для функционирования сайта, есть другие, значения которых на первых стадиях нас устраивают.

4.3 Настройка MySQL

Базы данных по сути входят в отдельную категорию, отличную от общего программирования с их механизмами хранения данных, репликациями и формами передачи данных, не касаясь реляционной алгебры – *реляционная алгебра* базируется на теории множеств и является основой логики работы баз данных.

В нашем случае проект не на столько сложный. Конфигурация БД (базы данных) происходит по следующему алгоритму: настройке подлежит файл `my.cnf` со всеми параметрами, создание самой базы данных с верной кодировкой, создание пользователя базы данных.

В файле `my.cnf`, который обычно находится по адресу `/etc/mysql/my.cnf` можно настроить достаточно важные параметры. Первостепенно необходимо определиться с типом движка базы данных или системой хранения данных в

MySQL 5 их две, отличия фундаментальны.

MyISAM – первый тип хранения данных разработанный для MySQL и присутствующий в ранних версиях. Основное достоинство на данный момент, скорость чтения, расход оперативной памяти минимальный. InnoDB – относительно не давно разработанный тип движка начиная с MySQL версии 3.23. Из основных преимуществ отказоустойчивость при записи в базу, т.е. если в один из моментов произошёл сбой, то InnoDB в отличии от MyISAM не допустит частичного записывания данных и тем самым нарушения работы приложения, так же как и потери данных. Восстановление базы на MyISAM после сбоя очень трудоёмкий процесс и чреват потерей данных. Однако в отличии от MyISAM, InnoDB сильно грузит жёсткие диски, для компенсации I/O (ввода-вывода) диска, многие ставят большой буфер, параметр `innodb_buffer_pool_size` порой больше 50% всей имеющейся памяти на сервере, потому что пропуски на жёстком диске недопустимы и это приведёт к зависанию системы.

Другие существенные различия механизмов хранения данных:

- поддержка внешних ключей – метод объединения записи в двух таблицах по определенным полям. Когда при обновлении поля в родительской таблице происходит определение и изменение поля в дочерней таблице. В MyISAM не поддерживается;

- файловое хранение таблиц – у MyISAM каждая таблица в отдельном файле, InnoDB данные хранятся в больших совместно используемых файлах;

- блокировка на уровне строк в InnoDB, позволяет обновлять строки параллельно;

- отказоустойчивость InnoDB, в случае ошибки в одном запросе, возвращается к предыдущему состоянию и только в случае выполнения всех запросов подтверждается транзакция;

- смертельный захват – процесс в многозадачной среде или СУБД, в следствие чего процессы остаются в состоянии ожидания ресурсов на неопределённый срок возможны только в InnoDB;

- полнотекстовый поиск – в InnoDB до версии MySQL 5.6.4;

- бинарное копирование таблиц – копирование таблиц между компьютерами разных архитектурах и соответственно разными операционными системами. В InnoDB не поддерживается.

Это не все различия и изменения происходят от версии к версии. На этом невозможно остановиться в настройке базы данных, так как можно получить ошибку слишком большой пакет или же очередь исчерпана при использовании приложения.

Рассматривая файл `my.cnf` по умолчанию имеет скудные настройки, и даже параметр `innodb_buffer_pool_size` не задан. Изначально без переопределения MySQL станет использовать InnoDB механизм хранения данных. Остановившись кратко по параметрам получим небольшой список:

- `key_buffer_size` – важный параметр для MyISAM. Так как механизм использует данный буфер для хранения данных в отличии от InnoDB использующий его только под индексы;

- `table_cache` – MyISAM помечает файлы таблиц как открытые при работе

с ними, для того чтобы не открывать таблицы слишком часто необходимо их запомнить. Данный параметр отвечает за запоминание открытых таблиц, размер кеша стоит увеличивать при большом количестве используемых одновременно таблиц;

- `thread_cache` – при соединении и разрыве создаётся или уничтожается поток, для сохранения состояний потоков используется данный параметр;

- `max_connections` – максимальное количество соединений к серверу, важный параметр, можно получить ошибку и отказ в доступе, например, при внесении данных большой формы, которую потому не захочется заполнять заново;

- `max_user_connections` – максимальное количество одновременных подключений для одного пользователя;

- `max_allowed_packet` – случай с чересчур большим пакетом, стандартная ошибка «Packet Too Large» необходимо изменить до размеров самого большого пакета.

Для отображения всех параметров необходимо войти на сервере в MySQL выполнив команду

```
mysql -u root -p
```

Где `u` – пользователь, в нашем случае суперпользователь, `p` – пароль. При использовании команды `SHOW VARIABLES`; внутри MySQL сервера получим вывод всех параметров им используемых.

```
mysql> SHOW VARIABLES;
```

Variable_name	Value
auto_increment_increment	1
auto_increment_offset	1
autocommit	ON
automatic_sp_privileges	ON
back_log	50
basedir	/home/bin/mysql-5.5
big_tables	OFF
binlog_cache_size	32768
binlog_direct_non_transactional_updates	OFF
binlog_format	STATEMENT
binlog_stmt_cache_size	32768
bulk_insert_buffer_size	8388608
...	
max_allowed_packet	4194304
max_binlog_cache_size	18446744073709547520
max_binlog_size	1073741824

Неправильная конфигурация MySQL может привести к потере данных, прерыванию работы приложения, низкой скорости обработки запросов. Изменить механизмы хранения данных на «лету» невозможно и таблицы нужно конвертировать, что проблематично при серьёзных размерах баз данных. Это далеко не все параметры обычной конфигурации.

При создании базы данных необходимо указать верную кодировку данных, сопряжённую с кодировкой приложения. Наша ИС использует так называемую универсальную кодировку (UTF-8 Unicode Transformation Format, 8-bit) которая

поддерживает почти все языки в том числе и Русский.

Для создания и управления базами данных можно воспользоваться стандартными клиентами, как phpMyAdmin, Navicat и т.п. В каждую базу данных можно добавить одного или нескольких пользователей для доступа приложений или каких-либо программ.

4.4 Настройка Apache

Разработка сего надёжного веб-сервиса принадлежит Робу Маккулу и создавался он для суперкомпьютеров в Университете Иллинойса США (NCSA). В 1994 году Роб выложил первый веб-сервер, который так и называл – NCSA. Однако в этом же году разработки прекратились, так как Роб покинул университет. Однако проект не был забыт, узкий круг людей заинтересованный и знавший о разработке начала совместную работу над продуктом. Они начали делать доработки и нововведения для сервиса. Появилась новая группа Apache Group, разработавшая Apache-сервис в апреле 1995 года на основе NCSA Server 1.3. На сегодняшний день это самый популярный сервис для протокола HTTP.

Одна из последних серьёзных доработок, это непоследовательная обработка запросов. Надо заметить, что делать параллельные запросы в последовательно обрабатываемом коде до сих пор проблематично. Технически Keep-Alive это метод для повторного использования соединения. Когда подключение создано, клиент отправляет некоторые установочные параметры на сервер. С одной стороны, на открытие соединения не уходит много времени, однако на сегодняшний день сайты имеют колоссальное количество картинок, файлов стилей, файлов JS которые загружаются разрозненно. Создавать на каждый файл новое соединение достаточно ресурсоёмко, с этим и борется Keep-Alive, данный метод позволяет использовать одно подключение для всех подгружаемых файлов одного пользователя.

Даже здесь не всё так идеально в действительности, если у нас к примеру, есть index файл и 20 CSS файлов, подгружаемых вместе с ним то, одно соединение загружает 3 файла, следовательно, для полной загрузки необходимо 7 соединений. Что в принципе достаточно серьёзно ускоряет процесс, до 300 мс выигрыша (при пинге 10-20 мс), для статической страницы серьёзная прибавка в скорости.

Одна из серьёзных проблем этого метода ресурсоёмкость отдельного процесса. Так как один процесс использует достаточно много оперативной памяти и ожидает повторного соединения, при отсутствии достаточного количества свободных ресурсов может произойти повисание сервиса. Правильная настройка очень важна.

Для настройки Keep-Alive на нашей виртуальной машине необходимо отредактировать файл /etc/apache2/apache2.conf:

- KeepAlive – флаг включения/отключения (On/Off);
- MaxKeepAliveRequests – так как один клиент может открыть несколько соединений, последние браузеры открывают по 6, то реальная цифра пользователей – это цифра разрешённых соединений делёная на 6;

– `KeepAliveTimeout` – количество секунд, которые сервис использует для ожидания соединения после последнего запроса. По умолчанию 15 секунд, если необходимо сэкономить память ставим меньше. Нужно помнить, что при слишком маленьких значениях на открытие и закрытие соединений расходуется процессорное время. Правильно поставить дополнительную память, сейчас очень недорогостоящая составляющая и выставить хорошие таймауты;

На этом нельзя завершить настройки Apache, как указывалось выше для обработки запросов для одного из доменов, ссылающегося на сервер, для домена как минимум нужно определить корневую директорию где располагаются файлы сайта см. раздел 1.3.

Отдельно стоит вынести другой конф. файл `/etc/apache2/mods-available/mpm_prefork.conf`

Где присутствует следующие директивы:

- `StartServers` – дочерние процессы при запуске;
- `MinSpareServers` – количество простаивающих процессов, минимум;
- `MaxSpareServers` – количество простаивающих процессов, максимум;
- `MaxRequestWorkers` – количество подключений к сервису, необходимо установить хорошее значение иначе произойдет отказ в обслуживании;
- `MaxConnectionsPerChild` – количество подзапросов за одно соединение, достигнув этого значения процесс перезапускается, защищает от атак на отказ сервиса. Не рекомендуется указывать значения меньше 1000 (0 ограничений нет).

Сам сервис имеет модульную структуру, модули можно подключать и отключать по необходимости. Стоит отметить значение `.htaccess` файла в котором можно переписывать запросы из адресной строки. Обычно используется сайтами в целях безопасности, указание ложных путей и создания читабельных названий страниц. Файл `.htaccess` размещают в корневой директории сайта. При помощи него возможно даже создать элементы логики работы сайт, особенно это используется при кешировании страниц, когда создаются перенаправления на статические страницы нежели динамические, что экономит ресурсы.

После всех конфигураций необходимо перезапускать сервис для того чтобы они вступили в силу. Делать это лучше проверив синтаксис

```
service httpd configtest
```

```
получаем ответ
```

```
Syntax OK
```

```
делаем рестарт
```

```
service httpd restart
```

```
Stopping httpd: [ OK ]
```

```
Starting httpd: [ OK ]
```

4.5 Резервное копирование проекта

Резервное копирование очень важный элемент при разработке сайта, так как ошибки, потеря данных это неотъемлемая часть при работе над проектом особенно команды разработчиков. На системах типа Unix предусмотрены свои

собственные утилиты для резервного копирования. При правильном резервном копировании сохраняются метаданные о файлах: как права доступа, владелец файла (пользователь), группы, дата последнего редактирования.

Важно отметить, что полностью сохранять данные сайта при изменении за время работы нескольких файлов нецелесообразно. Существует инкрементное резервное копирование, при котором сохраняются только те файлы, которые были изменены со времени последнего резервного копирования.

Наиболее простым способом локального резервного копирования является архивация данных с помощью `tar`. Обладает всеми преимуществами как инкрементное резервное копирование (оперирование отдельными файлами в архиве добавление/удаление), извлечение отдельных файлов из архива.

Для создания резервной копии следует воспользоваться командой

```
tar -cvzpf /var/www/school /var/www/school.tar.gz,
```

где `c` – создание архива, `v` – вывод процесса резервного копирования на экран, `z` – сжатие данных, `f` – сообщает что у нас файловый архив, флаг `p` – служит для сохранения владельцев и прав доступа файлов. При использовании инкрементного резервного копирования архивация выполняется с параметром `listed-incremental` – который указывает на файл метаданных, в котором хранится информация о состоянии архива. В случае, если файла с метаданными не существует, то при выполнении команды создается полный архив, а также новый файл метаданных, в последствии используемый для создания инкрементных резервных копий. Флаги утилиты, так же, как и любой другой Unix команды можно просмотреть набрав `man tar`.

Рассмотрим также ещё одну из основных утилит для сохранения данных сайта и не только на Unix системах. Утилита `rsync` на первый взгляд сложна в использование, однако, позволяет настроить резервное копирование достаточно тонко.

```
rsync -chavzP --stats -e 'ssh -p 1419' --exclude '/cache/'  
root@82.146.44.167:/var/www/school/ /var/www/school/
```

Используя сложный алгоритм `rsync` может передавать только изменения в файлах, при этом для выявления изменений ей не нужно сравнивать файлы. Передача данных происходит по протоколу SSH, поддерживающий шифрование. При этом `dir1` всегда откуда копируем, а `dir2` куда, на любом из серверов как отдающем, так и принимающем, для директории отдельно можно указать IP и пользователя. Флаги используются в утилите для создания резервной копии школьного сайта выполняют следующие предназначение:

- `c` или `checksum` – проверяет контрольные суммы файлов для синхронизации, нужно учесть, что первая резервная копия делается полностью;
- `h` – переводит указание размеров файлов в привычные МБ и ГБ;
- `v` – увеличивает уровень подробностей при выводе процесса на экран;
- `a` – архивный режим, не используется без `z` тип сжатия `gzip`, на медленных линиях эффективный способ ускорения передачи данных;
- `P` – показывает передачу данных для отдельно файла;

– stats – статистика при завершении процесса резервного копирования, сколько файлов передано, общий размер, сколь всего файлов и т.п. рисунок 12;

```
Number of files: 5771
Number of files transferred: 0
Total file size: 54130811630 bytes
Total transferred file size: 0 bytes
Literal data: 0 bytes
Matched data: 0 bytes
File list size: 121657
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 139660
Total bytes received: 18002

sent 139660 bytes received 18002 bytes 4095.12 bytes/sec
total size is 54130811630 speedup is 343334.55
Total time: 0:00:38.639000
```

Рисунок 12 – Статистика при завершении процесса резервного копирования

– e – очень полезное свойство утилиты резервного копирования, позволяющие выбрать альтернативную программу для соединения между удаленной и локальной копиями, программа или удалённая оболочка тоже может быть со своими флагами. Для SSH правильно изменить порт по умолчанию в целях безопасности, соответственно изменённый порт нужно указать при соединении;

– exclude – исключает директорию из синхронизации.

Есть ещё полезные свойства (флаги) утилиты, как delete – удаление файлов на удалённом сервере, с учётом того, что они были удалены на локальном сервере.

Резервное копирование файлов гораздо проще, нежели резервное копирование базы данных. Полноценной программы по синхронизации данных за исключением репликаций сейчас не существует. Репликации одно из основных свойств MySQL невозможно отнести к резервному копированию, так как все копии динамические.

Один из самых простых способов для небольших баз создать так называемый дамп базы при помощи Unix утилиты *mysqldump*

`mysqldump -u Пользователь -pПароль БАЗА > /path/to/file/dump.sql`

На выходе получаем текстовый файл всей базы данных, использовать *mysqldump* без дополнительных флагов опрометчиво. Особенно если версии MySQL серверов разные, т.е. база восстанавливается на другой машине. При резервном копировании стоит добавить флаг *opt* (add-drop-table, add-locks, all, quick, extended-insert, lock-tables, и disable-keys) указывает все перечисленные флаги вместе взятые. Необходимо отметить что, например, флаг *quick* – меняет поведение утилиты и база сохраняется построчно, а не читается целиком в память, а затем пишется на диск. Так как все утилиты Unix низкоуровневые и на прямую работают с «железом» стоит ознакомиться со всеми свойствами утилиты перед её применением.

Существуют и другие способы, наиболее подходящий способ для нашего

проекта воспользоваться клиентом MySQL, например, Navicat и скопировать базу в другой репозиторий при помощи свойства программы «перемещения данных». Так как клиент пользуется настройками MySQL серверов, то ошибиться сложно и данные останутся в сохранности.

Так как базы бывают большими по 100 ГБ и больше, не всегда удобно пользоваться приведёнными выше методами для сохранения базы, так как инкрементного резервного копирования тут нет, а времени занимает это много. Более продвинутый способ innobackupex – это отдельная нестандартная утилита, устанавливаемая в Unix-се обычным способом. Её преимущество в том, что она сохраняет файлы таблиц и может их восстановить, при этом доступны все операции по сжатию файлов и т.п. Происходит это гораздо быстрее обычных способов сохранения баз данных. В том числе есть поддержка инкрементного резервного копирования

```
innobackupex --incremental /data/backups --incremental-basedir=BASEDIR
```

Которое основывается на поддержке механизмом InnoDB логирования порядкового номера версии базы. Получается деление на страницы, утилита может восстановить процесс обновления базы и сохранить изменения. Возможно также сохранения базы с указанием инкремента или числа порядкового номера

```
innobackupex --incremental /data/backups --incremental-lsn=1291135
```

Восстановление же базы в этом случае немного сложнее, нужно подготовить полный бэкап, записав в него все изменения, произошедшие за всё время с первого резервного копирования и только затем обычным способом восстанавливать базу данных. Делается это следующим образом

```
innobackupex --apply-log --redo-only BASE-DIR --incremental-dir=INCREMENTAL-DIR-1
```

Флаг redo-only – применяется ко всем обновлениям первичного бэкапа за исключением последнего. Понятно, что первичная версия сохранённой базы нарушится и желательно перед всеми действиями её повторно сохранить. Структура дерева расположения инкрементных копий по дате проста и здесь сложно ошибиться. В целом утилита удобна при разработках, особенно если достаточно часто делать «снимки» базы.

4.6 Firewall или базовые настройки безопасности

Оставлять проект без первичных мер безопасности не рекомендуется. У Ubuntu ОС достаточно просто можно установить и настроить Advanced Policy Firewall (APF).

Для защиты от внешних воздействий у сервера необходимо настроить iptables. Настраивать iptables написанием большого количества правил слишком долго и трудоёмко при этом надо учесть автозагрузку при сбоях и перезагрузках. APF готовое решение, которое тоже требует рассмотрения. Простота в установке и изученность позволяют справиться с настройками относительно быстро. APF генерирует список правил и добавляет их в iptables, заменяя существующие и создавая демон. Поэтому последующие правила в таблицы следует добавлять только через firewall. От серьёзных нападений он не защитит, однако серьёзно

усложнит жизнь злоумышленникам.

Установка APF на Ubuntu

`apt-cache search apf`

В случае, если название пакета достоверно неизвестно, необходимо воспользоваться поиском, приведённая команда выше выдаст нам вывод вида:

`apf-client` - Client for Active Port Forwarding

`apf-firewall` - Easy iptables based firewall system

`apf-server` - Server for Active Port Forwarding

Устанавливаем firewall

`apt-get install apf-firewall`

Получаем сообщение, что для запуска демона требуется изменить переменную `RUN` с `no` на `yes` в файле `/etc/default/apf-firewall`. Не стоит сразу менять параметр не разобравшись, так как можно заблокировать себе доступ.

* `apf-firewall` disabled, please adjust the configuration to your needs

* and then set `RUN` to 'yes' in `/etc/default/apf-firewall` to enable it.

Открываем конфигурацию, файл APF

`pico -w /etc/apf-firewall/conf.apf`

В этом файле нужно указать порты, которые вы хотите оставить открытыми. Правим следующие переменные:

`IG_TCP_CPORTS="21,22,25,53,80,443,110,143"` – в этой переменной через запятую в кавычках описываются входящие tcp порты.

`IG_UDP_CPORTS="20,21,53,123"` – в этой аналогично исходящие tcp порты.

Тоже самое для `udp`:

`IG_UDP_CPORTS="21,22,25,53,80,443,110,143"`

`EG_UDP_CPORTS="20,21,53,123"`

Сохраняем файл и пробуем запустить firewall:

`apf -s`

Возможно, если что-то не так, то после запуска `firewall` заблокируется доступ к серверу. Потенциально, если директива `DEVEL_MODE` не изменена `firewall` отключится через 5 минут. Если всё в порядке, `firewall` работает нормально и доступ к серверу вы не потеряли, то следует изменить значение переменной `DEVEL_MODE` на 0, в конфигурационном файле. По умолчанию у неё значение 1, это режим APF для запуска в пятиминутном интервале. Всё, APF установлен.

Так же в каталоге `/etc/apf-firewall` есть множество конфигурационных файлов для более тонкой настройки `firewall`. Внутри почти всех файлов есть комментарий для чего данный конфиг. и описан синтаксиса. Можно, например, открыть доступ к определённому порту только с определённого IP или добавить обычное правило `iptables`. В общем APF очень гибок в настройке и понятен в использовании.

Часто APF используют для защиты от сопутствующих атак, т.е. не только использованию незакрытых портов или их взлому. Например, Bruteforce – любая задача может быть решена полным перебором. При этом, даже если вычисление целевой функции от каждого конкретного возможного решения задачи может

быть осуществлено за полиномиальное время, в зависимости от количества всех возможных решений полный перебор может потребовать экспоненциального времени работы.

Атаки по подбору паролей к серверу и приложениям очень часты. Для защиты стандартных портов от взлома совместно с APF необходимо установить Brute Force Detection (BFD). Данная утилита вносит в чёрный список IP с повторяющимися запросами на доступ и ошибочными паролями. По умолчанию поддерживаются стандартные протоколы как SSH и т.п.

Изменение стандартных портов, если протокол позволяет изменить порт, через который происходит соединение с внешним миром, стоит использовать. Так же, как и использование ключей вместо паролей, при поддержке протоколом, тоже стоит перейти на ключи доступа.

ЗАКЛЮЧЕНИЕ

Данная работа ставит перед собой цель в описание и разработке систем по созданию автоматизированных обработчиков информации или информационных систем.

В работе была рассмотрена типичная жизненная ситуация по созданию школьного сайта с расписанием, дневниками и пользователями. Приводились примеры используемых технологий, широко рассматривались методы реализации привычных нам вещей в повседневности в электронном виде.

Необходимо понимать, что всё разнообразие по созданию информационных систем и их техническая реализация не вместится и в нескольких томах.

Была рассмотрена достаточно интересная и своеобразная система Drupal. В ней реализован отличающийся от привычного ООП подход программирования. Системе уже более 15 лет. Ключевой момент в развитии системы произошёл в 2007 году с появлением 6 версии и наиболее популярных модулей как Views, ССК и т.п.

Нужно понимать, что в разработке системы принимала участия корпорация IBM. Первичная документация выкладывалась именно на сайтах этой корпорации. Мало того, есть мало известные факты, например, для отображения загрузки файлов была разработана библиотека для PHP, именно создатели Drupal были инициаторами, т.к. на момент разработки системы PHP был сильно ограничен и не имел даже объектное ориентированного подхода.

В работе также затрагивался, вскользь, вопрос о экономии времени и сил, современные системы позволяют устанавливать множество необходимых составляющих из пакета. Не далеки те времена, когда для установки PHP на сервере нужна была ручная настройка конф. файла с прописыванием всех собранных библиотек вручную и двойная компиляция самого языка.

Хочется отметить, что современность стремиться к интерфейсам, в том числе и программируемым, что никак не облегчает жизнь программисту, но ускоряет создание приложений.

Работа косвенно затрагивает основы защиты информации при распределении прав по ролям и создание сценариев поведения пользователей.

Глубоко рассмотрена первичная конфигурация сервера после установки пакета LAMP. Значение правильной конфигурации составляющих пакета сложно недооценить, ведь неверные значения параметров конфигурации могут привести к отказу в обслуживании.

Рассмотрен основной метод резервного копирования данных на Unix системах, как локального резервного копирования, так и на удалённые сервера.

В работе не были рассмотрены программы контроля версий, которые при работе в команде являются неотъемлемой частью рабочего инструментария. Соответственно не указывались и методики совместного программирования. Также не затрагивались системы отслеживания багов и т.п. вещи.

Проект всегда может быть продолжен.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Колисниченко, Д.Н. Linux. От новичка к профессионалу / Д.Н. Колисниченко; 4-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2012. – 704 с.
2. Мелансон, Б. Профессиональная разработка сайтов на Drupal 7 / Б. Мелансон, Д. Нордин, Ж. Луиси. – Санкт-Петербург : Питер, 2013. – 688 с.
3. Негус, К. Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд / К. Негус; 2-е изд. – Санкт-Петербург : Питер, 2014. – 384 с.
4. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон; 4-е изд. – Санкт-Петербург : Питер, 2016. – 768 с.
5. Unix и Linux: руководство системного администратора / Э. Немет, Г. Снайдер, Т. Хейн, Б. Уэйли; 4-е изд., перевод с английского – Москва : ООО «И.Д. Вильямс», 2012. – 1312 с.
6. Документация по системе управления сайтами Drupal для разработчиков от 2017 [Электронный ресурс]// Справочная система. – режим доступа: <https://www.drupal.org/documentation>
7. Янк, К. PHP и MySQL. От новичка к профессионалу / К. Янк. – Москва : Эксмо, 2013. – 384 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Техническое задание

Назначение сайта (цель создания сайта):

Сайт разрабатывается для использования учениками, родителями учеников и учителями. В разработке используется общедоступная бесплатная система Drupal. Создание сайта имеет целью автоматизацию системы обучения, с этой целью создаются электронные дневники, расписание занятий, электронные тесты.

Локальная операционная система разработки:

Ubuntu 12.04 LTS (32 бита)

Язык сайта:

Русский

Основные требования к сайту:

При определении дизайна сайта в качестве наглядных примеров можно использовать сайты соответствующей тематики. Каждая страница сайта должна содержать логотип и название/номер школы. Логотип должен по гиперссылке переводить на главную страницу. Внизу на каждой странице сайта должна быть указана контактная информация или форма обратной связи.

Основные страницы и разделы сайта

а) Главная страница, сайт загружается с этой страницы:

- 1) общая информация, логотип, общее описание сайта, ссылки на основные страницы (меню);
- 2) блок, вход для пользователей;
- 3) контактная информация;

б) Расписание, страница включает следующую информацию: дату и время проведения предмета для класса учащихся;

в) Дневник с заданиями и полями для оценок:

- 1) требования к заданиям: проверяющие, оценка, замечания;
- 2) поля для предметов: время, оценка.

Минимальное разрешение монитора

Минимальное разрешение монитора, в котором будет просматриваться сайт 1024 x 800 пикселей. При указанном разрешении должна быть обеспечена возможность просмотра страниц сайта без горизонтальной прокрутки браузера.

Браузеры

Браузеры, с помощью которых будет просматриваться сайт:

– Microsoft Internet Explorer 6.0 и выше;

- Chrome;
- Mozilla Firefox;
- Opera.

Имеющиеся графические материалы

Лого сайта:



Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Информационные системы

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой ИС

_____ С.А. Виденин

подпись инициалы, фамилия

« ____ » _____ 20 ____ г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме _____ бакалаврской работы
бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту: Дубинчику Артёму Станиславовичу

фамилия, имя, отчество

Группа: ЗКИ12-07Б-1 Направление (специальность): 09.03.01

номер

код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы: Автоматизированная система обучения

Утверждена приказом по университету № 2904/с от 07.03.2017 г.

Руководитель ВКР: И.А. Ефремов, к.ф.-м.н., доцент кафедры

инициалы, фамилия, должность, ученое звание и место работы

«Информационные системы» ИКИТ

Исходные данные для ВКР: список требований к разрабатываемой системе, система реализации, методические указания.

Перечень разделов ВКР: ведение, общие сведения и предустановка системы, реализация первой стадии разработки системы обучения, реализация проекта, дополнительные сведения по конфигурации и защите проекта, заключение.

Перечень графического материала: презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР

подпись

И.А. Ефремов

инициалы и фамилия

Задание принял к исполнению

подпись

А.С. Дубинчик

инициалы и фамилия студента

« 29 » марта 2017 г.